

Automated Hardware Design via Evolutionary Search

Jason D. Lohn, Silvano P. Colombano

The goal of this research is to investigate the application of evolutionary search to the process of automated engineering design. Evolutionary search techniques involve the simulation of Darwinian mechanisms by computer algorithms. In recent years, such techniques have attracted much attention because they are able to tackle a wide variety of difficult problems and frequently produce acceptable solutions. The results obtained are usually functional, often surprising, and typically "messy" because the algorithms are told to concentrate on the overriding objective and not elegance or simplicity.

Automation of engineering design has numerous advantages. First, faster design cycles translate into time and, hence, cost savings. Second, automated design techniques can be made to scale well and hence better deal with increasing amounts of design complexity. Third, design quality can increase because design properties can be specified *a priori*. For example, size and weight specifications of a device, smaller and lighter than the best known design, might be optimized by the automated design technique. The domain of electronic circuit design is an advantageous platform in which to study automated design techniques because it is a rich design space that is well understood, permitting human-created designs to be compared to machine-generated designs.

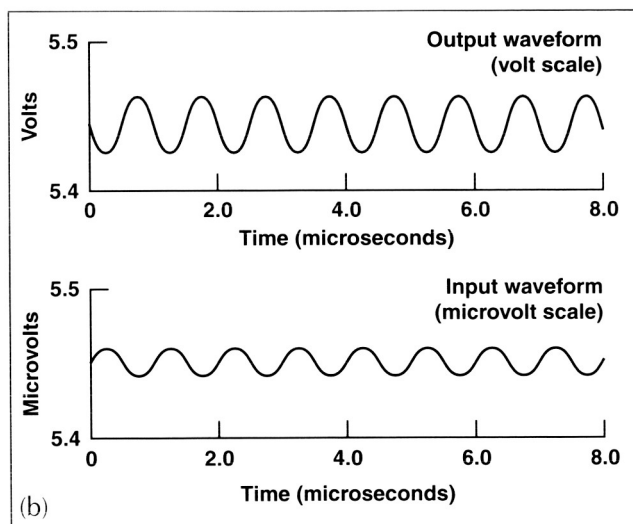
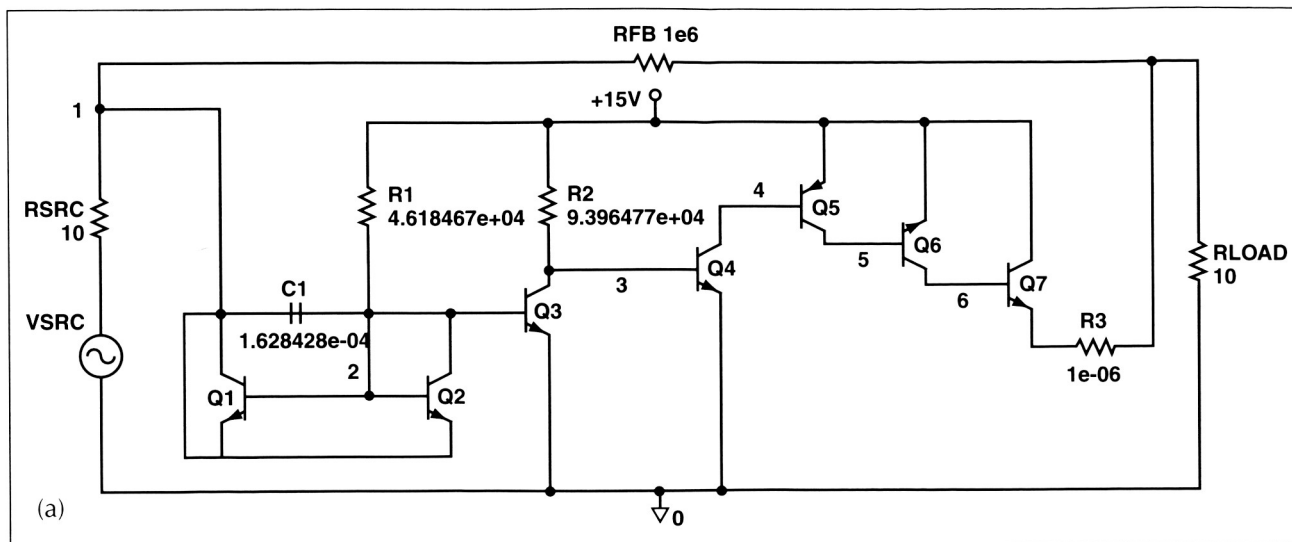
The overall goal of the evolutionary algorithms developed for circuit design was to automatically produce high-level integrated electronic circuit designs whose properties permit physical implementation in silicon. This process entailed designing an effective evolutionary algorithm and solving a difficult multiobjective optimization problem. FY99 saw many accomplishments in this effort.

First, the linear representation of circuits—a circuit-constructing programming language—was shown to be effective when used in an evolutionary algorithm. Transistor-placing language constructs were added since transistors are ubiquitous in analog integrated circuit design. Second, a technique for using coevolutionary search was shown to be better than manually encoded fitness schedules and as effective as using a single fitness function. Third, a 32-node parallel supercomputer, tailored for evolutionary algorithms, was obtained and deployed for automated circuit design. In addition, the parallel algorithm software was significantly enhanced to improve performance.

Coevolutionary search works by pitting two populations against each other, much like two teams of chess players, where all players on team A play all players on team B in a sequence of matches. The coevolution can be set up as a competition, as in the chess game metaphor, as a cooperation, or as a hybrid of the two. For circuit design, a competitive coevolution was used: a population of circuits was pitted against a population of circuit performance requirements. As soon as circuits were able to solve the most difficult performance requirements, those requirements were adjusted so that they would be more difficult to meet.

The system developed has successfully generated circuit designs for analog filters and amplifiers. Figure 1 shows an evolved 85-decibel (dB) amplifier (a circuit that outputs a voltage that is approximately 18,000 times the input voltage), its performance characteristics, and its genetic representation. This design is electrically well behaved, but is not yet implementable because of specifications that were not included in the fitness function.

Point of Contact: J. Lohn/S. Colombano
(650) 604-5138/4380
jlohn@ptolemy.arc.nasa.gov
scolombano@mail.arc.nasa.gov



```

transistor(N, ACTIVE_NODE, NEW_NODE, INPUT_NODE);
transistor(N, BASE, ACTIVE_NODE, PREVIOUS_NODE);
resistor_cast_to_ps(4.618467+04);
capacitor_cast_to_input(1.628423E-04);
transistor(N, NEW_NODE, ACTIVE_NODE, GROUND_NODE);
resistor_cast_to_ps(9.396477E+04);
transistor(N, NEW_NODE, ACTIVE_NODE, GROUND_NODE);
transistor(P, NEW_NODE, ACTIVE_NODE, PS_NODE);
transistor(N, NEW_NODE, ACTIVE_NODE, PS_NODE);

```

(c)

Fig. 1. (a) Circuit schematic of evolved 85-dB amplifier. The design of this circuit—topology, component types, component values—was produced automatically by the parallel evolutionary design algorithm. (b) Small-signal behavior of the evolved amplifier. Traces show the output is inverted and amplified. (c) Genetic representation of the evolved amplifier. The evolved amplifier is designed using a special circuit design programming language devised by the authors.